PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPLICANT(S):	Altman et al.	DOCKET: YOR920000844US1 (8728-473)
SERIAL NO.:	09/845,693	GROUP ART UNIT: 2183
FILED:	April 30, 2001	EXAMINER: Huisman, David J.
FOR:	**SYSTEM AND METHOD INCLUDING DISTRIBUTED INSTRUCTION BUFFERS HOLDING A SECOND INSTRUCTION FORM**	

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

In response to the Final Office Action dated June 23, 2005, finally rejecting Claims 1, 2, 5-9, 11, 13 and 16-19 under 35 U.S.C. §102(b) and Claims 3, 10, 12, 15, 20 and 21 under 35 U.S.C. §103(a). Applicant appeals pursuant to the Notice of Appeal filed on August 22, 2005 and submits this appeal brief.

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Mail Stop Appeal-Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 24, 2005.

Dated: October 24, 2005

Melithza Rodriguez

TABLE OF CONTENTS

1. Real Party in Interest

The real party in interest is International Business Machines Corporation, the assignee of the entire right, title, and interest in and to the subject application by virtue of an assignment of record.

2. Related Appeals and Interferences

None.

3. Status of Claims

Claims 1-3, 5-13 and 15-21 are pending, stand rejected, and are under appeal.

Claims 4 and 14 have been cancelled.

A copy of the Claims as pending is presented in the Appendix.

4. Status of Amendments

Claims 2, 5, 10-12, 15, 18, 20 and 21 were amended by the Amendment under 37 C.F.R. §1.111, filed February 19, 2004. This Amendment was entered.

Claims 1, 11, 19 and 21 were amended by the Amended under 37 C.F.R. §1.116, filed July 14, 2005. By the amendment, claims 4 and 14 were cancelled. This Amendment was entered by way of a Request for Continued Examination, filed November 8, 2004.

Claims 1, 2, 7, 11, 13 and 20 were amended by the Amendment under 37 C.F.R. §1.111, filed April 25, 2005. This Amendment was entered.

5. Summary of Claimed Subject Matter

A processor includes a decoder for a primary instruction form stored in the primary instruction cache or memory. The processor also includes hardware for handling an alternate form of the instruction set stored in local predecoded instruction buffers. The alternate form of the instruction is generated by a compiler.

Referring to Claim 1, a computer-implemented method for processing a first instruction form and a second instruction form of an instruction set in a processor comprises providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form, wherein the first instruction form are decoded by a decoder in an execution pipeline and the second instruction form are predecoded by a compiler (see page 6, lines 6-14). The method includes storing the plurality of instructions of the second form in a plurality of buffers proximate to a plurality of execution units (see page 9, lines 14-16 and page 10, lines 8-15), and executing at least one instruction of the first instruction form in response to a first counter (see page 11, lines 12-13). The method further includes executing at least one instruction of the second instruction form in response to at least a second counter, wherein the second counter is invoked by a branch instruction of the first instruction form, (see page

11, line 13 to page 12 line 2) wherein the step of executing at least one instruction of the second instruction form further comprises the steps of de-gating a plurality of execution queues storing the plurality of instructions of the first instruction form (see page 12, lines 3-5), and pausing a fetching of the first instruction form from a memory (see page 12, lines 5-6).

Referring to Claim 11, a processor for processing a program of instructions comprising instructions of a first instruction form and a second instruction form comprises a plurality of execution units for receiving instructions (see Figure 3, elements 301-304 and page 10, lines 8-9), and a branch unit connected to an instruction fetch unit for the first instruction form (see Figure 3, element 305 and page 11, line 12) and a sequencer for the second instruction form, wherein the sequencer controls a plurality of gates connected between a plurality of execution queues for storing decoded instructions of the first instruction form and the plurality of execution units (see Figure 3, element 325 and page 11, line 18 to page 12 line 11). The processor includes a decode unit for decoding instructions of the first instruction form into control signals for the execution units (see Figure 3, element 323 and page 18, line 22 to page 19, line 1), and a plurality of buffers, proximate to the execution units, for storing predecoded instructions of the second instruction form (see Figure 3, elements 306-310 and page

9, line 18-19).

Referring now to Claim 21, a processor for processing a first instruction form and a second instruction form of an instruction set comprises a plurality of execution units for receiving instructions (see Figure 3, elements 301-304 and page 10, lines 8-9), and a branch unit connected to an instruction fetch unit for the first instruction form (see Figure 3, element 305 and page 11, line 12) and a sequencer for the second instruction form, wherein the branch unit switches the processor from the first instruction form to the second instruction form in response to a branch instruction of the first instruction form and switches the processor from the second instruction form to the first instruction form in response to a branch instruction of the second instruction form (see Figure 3, element 325 and page 11, line 8 to page 12, line 11). The processor comprises a decode unit adapted to decode instructions of the first instruction form into control signals for the execution units (see Figure 3, element 323 and page 18, line 22 to page 19, line 1), and an issue unit adapted to sequence decoded instructions of the first instruction form (see Figure 3, element 324 and page 11, line 21 to page 12 line 1). The processor includes a plurality of buffers, proximate to the execution units, for statically storing predecoded instructions of the second instruction form, wherein each execution unit is connected to a

corresponding buffer of the plurality of buffers (see Figure 3, elements 306-310 and page9, lines 18-19), and the sequencer, engaged by the branch unit, adapted to fetch the predecoded instructions and sequence the predecoded instructions of the second instruction form, wherein the sequencer is connected to a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates (see Figure 3, element 325 and page 11, line 18 to page 12, line 11).

6. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1, 2, 5-9, 11, 13 and 16-19 stand rejected under 35 U.S.C. 102(a) as being unpatentable over Parady (U.S. Patent No. 5,933,627).

B. Claim 3 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Parady (U.S. Patent No. 5,933,627).

C. Claims 10, 15 and 21 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Parady (U.S. Patent No. 5,933,627) in view of Hennessy and Patterson, "Computer Architecture - A Quantitative Approach, 2nd Edition," 1996.

D. Claim 12 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Parady (U.S. Patent No. 5,933,627) in view of Ball and Larus, "Efficient Path Profiling," 1996.

E. Claims 12 and 20 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Parady (U.S. Patent No. 5,933,627) in view of Lavi et al. (U.S. Patent No. 6,453,407).

7. **Argument**

A. The Claim Rejections Under 35 U.S.C. 102 Are Legally Deficient.

Under 35 U.S.C. §102, a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. The identical invention must be shown in as complete detail as is contained in the claim. See MPEP §2131.

i. Claims 1, 2, 5-9, 11, 13 and 16-19

It is respectfully submitted that at the very least, Parady is legally deficient to establish a case of anticipation against independent Claims 1 and 11.

Claim 1 claims, *inter alia*, "providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form, wherein the first instruction form are decoded by a decoder in an execution pipeline and the second instruction form are predecoded by a compiler." Claim 11 claims, *inter alia*, "a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the sequencer controls a plurality of gates connected between a plurality of execution queues for storing decoded instructions of the first instruction form and

the plurality of execution units."

Parady teaches a method and apparatus for switching between threads of a program (see Abstract). Parady does not teach "providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form, wherein the first instruction form are decoded by a decoder in an execution pipeline and the second instruction form are predecoded by a compiler" as claimed in Claim 1. Parady teaches an instruction cache and a decode unit through which *all* instructions are processed (see Figure 1 and col. 2, line 66-col. 3, line 10). The instructions of Parady are processed by the decoder, and for off-chip instructions, processed by a predecoder and the decoder. Thus, *all* instructions of Parady require decoding; no predecoded instructions are provided, much less instructions predecoded by a compiler. The instructions of Parady are not provided "predecoded by a compiler," essentially as claimed in claim 1. Thus, Parady fails to teach all the limitations of claim 1.

Further, in the Response to Arguments found in the Final Office Action dated June 22, 2005, the Examiner has compared the "decoding" of instructions written in a high-level language to instructions of a low-level language to "predecoded" instructions as claimed in Claim 1. Parady does not teach a

compiler that "predecodes"; compilers, such as that taught by Parady merely *translate* or *compile* high-level languages into low-level languages. Parady does not teach that a compiler generates predecoded instructions. The terms "decode" (or predecode) and "translate" or "compile" have definite meanings in the art and are not analogous or interchangeable.

In addition, referring to the Response to Arguments, the high-level instructions are not an instruction set that can be processed by a processor and thus are not analogous to either the first instruction form or the second instruction form as claimed in Claim 1. Thus, the rejection essentially states that: 1) low-level instructions are instructions of a first form (decoded by a decoder in an execution pipeline); *and* 2) low-level instructions are instructions of a second form (translated from the high-level instructions by a compiler). Clearly then, the low-level instructions cannot anticipate a program of instructions comprising the first instruction form *and* the second instruction form as claimed in Claim 1; for example, a program of instructions of Parady *only* includes the low-level instructions. Parady does not teach "providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form" as claimed in Claim 1.

Accordingly, for at least the reasons given above, Parady

fails to teach "providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form, wherein the first instruction form are decoded by a decoder in an execution pipeline and the second instruction form are predecoded by a compiler" as claimed in Claim 1.

Referring to Claim 11, Parady teaches a plurality of instruction buffers storing different threads of a program (see Figure 3). Parady does not teach "a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the sequencer controls a plurality of gates connected between a plurality of execution queues for storing decoded instructions of the first instruction form and the plurality of execution units" (emphasis added). The instruction buffers of Parady are controlled by thread switching logic (see element 112 of Figures 3). The thread switching logic is a pointer-based system for selecting a next thread from the instruction buffers (see col. 3, lines 57-65). The thread switching logic is not a gate connected between an execution queue for storing decoded instructions of the first instruction form and an execution unit, essentially as claimed in Claim 11. For example, the thread switching logic is not implemented *between* the instruction buffers and the execution unit (see for example, Figure 3 of Parady). Therefore, Parady

fails to teach all the limitations of Claim 11.

Claims 2 and 5-9 depend from Claim 1. Claims 13 and 16-19 depend from Claim 11. The dependent claims are believed to be allowable for at least the reasons given for Claims 1 and 11, respectively.

B. The Claim Rejections Under 35 U.S.C. 103 Are Legally Deficient.

In rejecting claims under 35 U.S.C. §103, the Examiner bears the initial burden of presenting a *prima facie* case of obviousness. In re Rijckaert, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a *prima facie* case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. In re Fine, 837 F.2d 1071, 1074 (Fed. Cir. 1988). A *prima facie* case of obviousness is established when the teachings of the prior art itself would appear to have suggested the claimed subject matter to one of ordinary skill in the art. In re Bell, 991 F.2d 781, 782 (Fed. Cir. 1993). If the Examiner fails to establish a *prima facie* case, the rejection is improper and must be overturned. In re Rijckaert, 9 F.3d at 1532 (citing In re Fine, 837 F.2d at 1074).

i. Claim 3

It is respectfully submitted that at the very least, the teachings of Parady are legally deficient to establish a *prima facie* case of obviousness against independent Claim 3.

Claim 3 depends from Claim 1. Claim 3 is believed to be allowable for at least the reasons given for Claim 1. Reconsideration of the rejection is respectfully requested.

ii. Claims 10, 15 and 21

It is respectfully submitted that at the very least, the combined teachings of Parady and Hennessy and Patterson are legally deficient to establish a *prima facie* case of obviousness against independent Claim 12 and 20.

Claim 10 depends from Claim 1. Claim 15 depends from claim 11. The dependent claims are believed to be allowable for at least the reasons given for Claims 1 and 11.

Claim 21 claims "the sequencer, engaged by the branch unit, adapted to fetch the predecoded instructions and sequence the predecoded instructions of the second instruction form, wherein the sequencer is connected to a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates.

Parady teaches a plurality of instruction buffers storing different threads of a program (see Figure 3). Parady does not teach "a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates" as claimed in Claim 21. The instruction buffers of Parady are controlled by thread switching logic (see element 112 of Figures 3). The thread switching logic is a pointer-based system for selecting a next thread from the instruction buffers (see col. 3, lines 57-65). The thread switching logic is not a gate connected between an execution queue for storing decoded instructions of the first instruction form and an execution unit, essentially as claimed in Claim 21. For example, the thread switching logic is not implemented *between* the instruction buffers and the execution unit (see for example, Figure 3 of Parady). Therefore, Parady fails to teach all the limitations of Claim 21.

Hennessy teaches reservation stations (see page 253, Figure 4.8). Hennessy does not teach or suggest "a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates" as claimed in Claim 21. The reservation stations of Hennessy are connected directly to the FP adders and

FP multipliers. Hennessy does not teach or suggest a gate disposed between the reservation stations and the FP adders and FP multipliers. Therefore, Hennessy fails to cure the deficiencies of Parady.

The combined teachings of Parady and Hennessy fail to teach or suggest "a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates" as claimed in Claim 21. Reconsideration of the rejection is respectfully requested.

iv. Claim 12

It is respectfully submitted that at the very least, the combined teachings of Parady and Ball and Larus are legally deficient to establish a *prima facie* case of obviousness against independent Claim 12 and 20.

Claim 12 depends from Claim 11. Claim 12 is believed to be allowable for at least the reasons given for Claim 1. Reconsideration of the rejection is respectfully requested.

iv. Claims 12 and 20

It is respectfully submitted that at the very least, the combined teachings of Parady and Lavi are legally deficient to

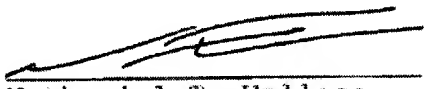
establish a *prima facie* case of obviousness against independent Claim 12 and 20.

Claim 12 depends from Claim 11. Claim 20 depends from Claim 11. The dependent claims are believed to be allowable for at least the reasons given for Claims 1 and 11, respectively. Reconsideration of the rejection is respectfully requested.

C. CONCLUSION

The claimed invention is not disclosed or suggested by the teachings of the applied prior art references, either alone or in combination. Moreover, the Examiner has failed to establish a case of anticipation under 35 U.S.C. §102 against independent Claims 1 and 11 over Parady or a *prima facie* case of obviousness under 35 U.S.C. §103 over Parady and Hennessy and Patterson with respect to Claim 21 for at least the reasons noted above. Claims 2, 3 and 5-10 depend from Claim 1. Claims 12, 13 and 15-20 depend from Claim 11. The dependent claims are believed to be allowable for at least the reasons given for Claims 1 and 11. Accordingly, it is respectfully requested that the Board overrule the rejections of Claims 1-3, 5-13 and 15-21.

Date: October 24, 2005

By: 

Nathaniel T. Wallace
Reg. No. 48,909
Attorney for Appellants

F. CHAU & ASSOCIATES, LLP
130 Woodbury Road
Woodbury, New York 11797
TEL: (516) 692-8888
FAX: (516) 692-8889

8. CLAIMS APPENDIX

What is claimed is:

1. A computer-implemented method for processing a first instruction form and a second instruction form of an instruction set in a processor comprising the steps of:

providing a program of instructions comprising a plurality of instructions of the first instruction form and a plurality of instructions of the second instruction form, wherein the first instruction form are decoded by a decoder in an execution pipeline and the second instruction form are predecoded by a compiler;

storing the plurality of instructions of the second form in a plurality of buffers proximate to a plurality of execution units;

executing at least one instruction of the first instruction form in response to a first counter; and

executing at least one instruction of the second instruction form in response to at least a second counter, wherein the second counter is invoked by a branch instruction of the first instruction form,

wherein the step of executing at least one instruction of the second instruction form further comprises the steps of

de-gating a plurality of execution queues storing the

plurality of instructions of the first instruction form, and

pausing a fetching of the first instruction form from a memory.

2. The method of claim 1, wherein the instructions of the first form and instructions of the second form are generated by a compiler, wherein instructions of the second form are statically loaded into the plurality of buffers as control signals ready for execution.

3. The method of claim 2, wherein instructions of the second form are more frequently executed than instructions of the first form.

5. The method of claim 1, wherein the step of executing at least one instruction of the second instruction form further comprises the steps of:

fetching at least one instruction of the second instruction form from a buffer of the plurality of buffers; and

sequencing the at least one instruction of the second instruction form to the execution units.

6. The method of claim 1, wherein the second instruction form is a logical subset of the first instruction form.

7. The method of claim 1, wherein the step of executing at least one instruction of the first instruction form further comprises the steps of:

fetching an instruction of the first form from a memory;

decoding the instruction; and

issuing the decoded instruction to at least one execution unit.

8. The method of claim 1, wherein a return to fetching of the first instruction form is signaled by a switch bit in a buffer of a branch unit storing instructions of the second form.

9. The method of claim 1, wherein a return to fetching of the first instruction form is signaled by a return instruction of the second instruction form stored in a buffer of a branch unit.

10. The method of claim 1, wherein each execution unit is associated with a different buffer of the plurality of buffers.

11. A processor for processing a program of instructions comprising instructions of a first instruction form and a second instruction form comprising:

a plurality of execution units for receiving instructions;

a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the sequencer controls a plurality of gates connected between a plurality of execution queues for storing decoded instructions of the first instruction form and the plurality of execution units;

a decode unit for decoding instructions of the first instruction form into control signals for the execution units; and

a plurality of buffers, proximate to the execution units, for storing predecoded instructions of the second instruction form.

12. The processor of claim 11, wherein the instructions of the first form and instructions of the second form are generated based on execution frequency, wherein instructions of the second form are executed more frequently than instructions of the first form.

13. The processor of claim 11, wherein the sequencer, engaged by the branch unit, addresses the predecoded instructions of the second instruction form stored in the buffers and sequences predecoded instructions of the second instruction form to the execution unit.

15. The processor of claim 11, wherein each execution unit is connected to a corresponding buffer of the plurality of buffers.

16. The processor of claim 11, wherein the branch unit switches the processor from the first instruction form to the second instruction form in response to a branch instruction of the first instruction form.

17. The processor of claim 11, wherein the branch unit switches the processor from the second instruction form to the first instruction form in response to a branch instruction of the second instruction form.

18. The processor of claim 11, wherein a switch bit in a buffer of the plurality of buffers connected to the branch unit signals the sequencer to stop fetching from the buffers and enables instruction fetching from a memory storing instructions of the first instruction form.

19. The processor of claim 11, wherein an execution bandwidth of the execution units is larger than a fetch/issue bandwidth of the first form.

20. The processor of claim 11, wherein the second instruction

form is a logical subset of the first instruction form, wherein the predecoded instructions of the second instruction form are statically stored in the plurality of buffers, and wherein the predecoded instructions of the second instruction form are control signals generated by a compiler and are not decoded during a runtime of the program.

21. A processor for processing a first instruction form and a second instruction form of an instruction set comprising:

a plurality of execution units for receiving instructions;

a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the branch unit switches the processor from the first instruction form to the second instruction form in response to a branch instruction of the first instruction form and switches the processor from the second instruction form to the first instruction form in response to a branch instruction of the second instruction form;

a decode unit adapted to decode instructions of the first instruction form into control signals for the execution units;

an issue unit adapted to sequence decoded instructions of the first instruction form;

a plurality of buffers, proximate to the execution units, for statically storing predecoded instructions of the second instruction form, wherein each execution unit is connected to a corresponding buffer of the plurality of buffers; and

the sequencer, engaged by the branch unit, adapted to fetch the predecoded instructions and sequence the predecoded instructions of the second instruction form, wherein the sequencer is connected to a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates.

1 FW - AF

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

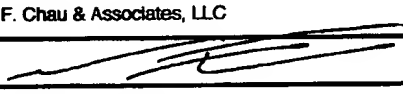
Application Number	09/845,693
Filing Date	April 30, 2001
First Named Inventor	Altman
Art Unit	2183
Examiner Name	Huisman, David J.
Attorney Docket Number	YOR920000844US1 (8728-473)

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please identify below):

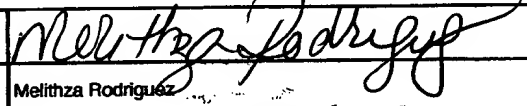
Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	F. Chau & Associates, LLC		
Signature			
Printed name	Nathaniel T. Wallace		
Date	October 24, 2005	Reg. No.	48,909

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Melithza Rodriguez	Date	October 24, 2005

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($)	500.00

Complete if Known	
Application Number	09/845,693
Filing Date	April 30, 2001
First Named Inventor	Altman
Examiner Name	Huisman, David J.
Art Unit	2183
Attorney Docket No.	YOR920000844US1 (8728-473)

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 50-0150 500510 Deposit Account Name: IBM/YORKTOWN HEIGHTS

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☐ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____ x	_____ =	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____ x	_____ =	_____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____ / 50 =	_____ (round up to a whole number) x	_____ =	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Appeal Brief	500.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 48,909	Telephone 516-692-8888
Name (Print/Type)	Nathaniel T. Wallace		Date October 24, 2005